in Computer Science, Amsterdam, pages 32–42. IEEE Computer Society Press, Los Alamitos, California, July 1991. Full paper to appear in *Information and Computation*.

[10] M. Kanovich. The multiplicative fragment of linear logic is NP-complete. Email Message, 1991.

[11] M. Kanovich. The multiplicative fragment of linear logic is NP-complete. Technical Report X-91-13, Institute for Language, Logic, and Information, June 1991.

[12] M. Kanovich. Horn programming in linear logic is NP-complete. In *Proc. 7-th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz*, pages 200–210. IEEE Computer Society Press, Los Alamitos, California, June 1992.

[13] S.C. Kleene. Permutability of inferences in Gentzen's calculi LK and LJ. *Memoirs of the AMS*, 1952.

[14] P. Lincoln and J. Mitchell. Operational aspects of linear lambda calculus. In *Proc. 7th IEEE Symp. on Logic in Computer Science*, 1992.

[15] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals Pure Appl. Logic*, 56:239–311, 1992. Special Volume dedicated to the memory of John Myhill abstract appeared in Proc. 31st IEEE Symp. on Foundations of Computer Science, 1990.

[16] P.D. Lincoln. *Computational Aspects of Linear Logic*. PhD thesis, Stanford University, 1992.

[17] R.A.G. Seely. Linear logic, *-autonomous categories, and cofree coalgebras. In: Categories in Computer Science and Logic, June 1989, 1989.

consider carefully the sources of exponential blowup identified by these suites of results (in this case, the splitting of contexts in applications of the $\otimes$ rule).

However, it is still essentially unknown how to harness the evident power of linear logic for useful purposes. Several interesting attempts have been made, including using linear logic as the basis for a logic programming language [9, 2], and as the basis for a functional programming language [1, 14]. The results given here have more direct impact on the logic programming approach, which is still in its infancy.

References

[1] S. Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 1991. Special Issue on the 1990 Workshop on Math. Found. Prog. Semantics. To appear.

[2] J.-M. Andreoli and R. Pareschi. Linear objects: Logical processes with built-in inheritance. In *Proc. 7-th International Conference on Logic Programming, Jerusalem*, May 1990.

[3] A. Avron. Some properties of linear logic proved by semantic methods. Technical Report 260/92, Eskenasy Institute of Computer Science, Tel-Aviv University, 1992.

[4] M. Barr. *-autonomous categories. In: Lecture Notes in Mathematics 752, Springer, 1979.

[5] G. Bellin. *Mechanizing Proof Theory: Resource-Aware Logics and Proof-Transformations to Extract Implicit Information.* PhD thesis, Stanford University, 1990.

[6] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Co., 1979.

[7] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[8] Jean-Yves Girard. Personal communication. April 1990.

[9] J.S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. In *Proc. 6-th Annual IEEE Symposium on Logic*

From an instance of Generalized-3-Partition, one may generate an instance of 3-Partition by adding $B' + 1$ to the size of each element of A'. The instance of 3-Partition then is asked with $B = 4B' + 3$, and the size of each element satisfies the condition $B/4 \leq s(a) \leq B/2$, since $B = 4B' + 3$, $s(a') < B'$, and $s(a) = s(a') + B' + 1$. By adding more than $B' + 1$ to the size each element, one can create instances of 3-Partition where elements are as close to $B/3$ as desired. Thus one could avoid the complications involved in "reshuffling" the groups of four and two elements above that arise with 432-Partition by using a properly restricted 3-Partition problem. The reshuffling only occurs for a with $s(a)$ very close to $B/4$ or $B/2$.

Another type of simplification can be achieved with other encodings of a 3-Partition problem. Consider the earlier encoding of 3-Partition in full multiplicative linear logic:

$$[(k \multimap c^{s(A_1)}) \otimes \cdots \otimes (k \multimap c^{s(A_{3m})}) \otimes (c^B \multimap j)^m] \multimap (k^3 \multimap j)^m$$

Constant-only encodings can be generated by replacing c by bottom, and k by $1^{\langle C \rangle}$ for some integer C. A value of C that is particularly interesting is $C = \sum_{a \in A} s(a)$. Although they are still polynomial, such encodings tend to be larger than the one advocated above, but result in somewhat less complicated proofs of soundness. The case of $C = 1$ is an incorrect encoding, and one may consider the "bottom only" encoding proved sound and complete above to be generated from the case $C = 0$.

3 Conclusion

We have demonstrated that simply evaluating expressions in *true*, *false*, *and*, and *or* in multiplicative linear logic ($\otimes$, $\parr$, 1, and $\bot$) is NP-complete. By conservativity results the NP-hardness of larger fragments of linear logic follow, although some of these results were known previously. These results comprise further dramatic evidence of the extreme expressive power of linear logic. Other results along these lines have previously shown that full propositional linear logic is undecidable, and there are natural fragments which are PSPACE-complete, EXPTIME-complete, and NP-complete.

Complexity results for fragments of linear logic indicate the difficulty of constructing efficient decision procedures for large fragments of linear logic. It may have been hoped previously that some "semantic" measure condition could be used to immediately decide constant-only expressions in linear logic. When constructing theorem provers for linear logic, one must

Thus, given any proof of $\Theta(\langle A, m, B, S\rangle)$, we first see that one may identify m branches, each of which is of the form $\vdash (1^{\langle X1\rangle} \otimes \bot), (1^{\langle X2\rangle} \otimes \bot), \cdots, (1^{\langle Xn\rangle} \otimes \bot), (\bot^B \wp 1^{\langle 3\rangle})$. From these m branches, we may identify m partitions of 4,3, or 2 elements of the associated 432-Partition problem. In other words, from any proof of the given sequent, one may construct a solution to the 432-partition problem. ∎

2.7 Main Result

From the preceding, we immediately achieve our stated result.

Theorem 2.4 (COMLL NP-COMPLETE) *The decision problem for constant-only multiplicative linear logic is* NP*-complete.*

Also, with an easy conservativity result, we find that this NP-Hardness proof suffices for multiplicative linear logic as well.

Theorem 2.5 (Conservativity) *Multiplicative linear logic is conservative over constant-only multiplicative linear logic.*

Proof. By induction on cut-free MLL proofs. ∎

2.8 Using 3-Partition Directly

Instead of using 432-Partition one could use 3-Partition directly with some simplifying assumptions.

One may also consider the following looser specification of 3-Partition, which we will call Generalized-3-Partition.

Instance:	Set A' of $3m$ elements, a bound $B' \in Z^+$, and a size $s(a') \in Z^+$ for each $a' \in A'$
Question:	Can A' be partitioned into m disjoint sets $A'_1, A'_2, \cdots, A'_m$ such that, for $1 \leq i \leq m$, $\sum_{a' \in A'_i} s(a') = B'$ such that each set contains exactly 3 elements from A'?

Generalized-3-Partition does not have a priori restrictions on the sizes of elements, but instead has an explicit specification that only partitions of three elements are allowed. One can immediately restrict $s(a')$ for all $a' \in A'$ to be $\leq B'$, for otherwise there is no solution, since all sizes are nonnegative.

Lemma 1.3, if there is a proof of this sequent, then there is a proof of $\vdash (1^{\langle S1\rangle} \otimes \bot), (1^{\langle S2\rangle} \otimes \bot), \cdots, (1^{\langle S3m\rangle} \otimes \bot), (\bot^B \wp 1^{\langle 3\rangle})^m$

We then perform complete induction on m.

If $m > 1$, the proof of this sequent must end in $\otimes$, since all formulas have main connective $\otimes$. We next show that the principal formula of that rule application must be $(\bot^B \wp 1^{\langle 3\rangle})^m$.

First, we note that each formula $(1^{\langle Sj\rangle} \otimes \bot)$ has measure $Sj - 1$. Since we are assuming $B > 8$, the initial conditions of the 432-Partition problem ensure that for all j, $Sj > 2$, and therefore $Sj - 1 > 1$. There is only one formula, $(\bot^B \wp 1^{\langle 3\rangle})^m$, with negative measure.

If we assume that one of the $(1^{\langle Si\rangle} \otimes \bot)$ formulas is principal in an application of $\otimes$, by Lemma 2.1, each hypothesis sequent must have measure one. In this case we have the following supposed proof for some Σ and Δ with the multiset union $\Sigma \bigcup \Delta \bigcup (1^{\langle Si\rangle} \otimes \bot)$ being equal to the conclusion:

$$\frac{\begin{array}{cc} \vdots & \vdots \\ \vdash \Sigma, \bot & \vdash \Delta, 1^{\langle Si\rangle} \end{array}}{\vdash (1^{\langle S1\rangle} \otimes \bot), (1^{\langle S2\rangle} \otimes \bot), \cdots, (1^{\langle S3m\rangle} \otimes \bot), (\bot^B \wp 1^{\langle 3\rangle})^m}\otimes$$

But $1^{\langle Si\rangle}$, which occurs in one hypothesis, has measure > 2. Therefore, the formula with negative measure, $(\bot^B \wp 1^{\langle 3\rangle})^m$, must occur in Δ. Now consider the other hypothesis, which must contain $\bot$, and other formulas Σ from the conclusion sequent. If any formulas of the form $(1^{\langle Sj\rangle} \otimes \bot)$ are included in Σ, the measure of that hypothesis is greater than 1. If no such formulas are included, then the sequent has measure 0. In either case, by Lemma 2.1, that sequent is not provable. Thus the assumption that one of the $(1^{\langle Sj\rangle} \otimes \bot)$ formulas is principal must be in error, and $(\bot^B \wp 1^{\langle 3\rangle})^m$ must be principal.

Thus if $m > 1$, the only possible next proof step is $\otimes$, with principal formula $(\bot^B \wp 1^{\langle 3\rangle})^m$. We may then focus on the case when $m = 1$. We claim that each such branch in the proof corresponds to one partition in the solution of the original 432-Partition problem. That is, we claim that when $m = 1$, we must be left with a sequent of the form:

$$\vdash (1^{\langle X1\rangle} \otimes \bot), (1^{\langle X2\rangle} \otimes \bot), \cdots, (1^{\langle Xn\rangle} \otimes \bot), (\bot^B \wp 1^{\langle 3\rangle})$$

Where the Xi are a subset of the Si. There are exactly $B + n - 1$ occurrences of $\otimes$ in this sequent, and $\sum_{1 \le i \le n} Xi + 3$ ones in this sequent. By Lemma 2.1, $(\sum_{1 \le i \le n} Xi + 3) - (B + n - 1) = 1$, or equivalently $\sum_{1 \le i \le n} Xi = B + n - 3$. This gives rise to an instance of 432-Partition.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash 1^{\langle Sx\rangle}, 1^{\langle Sy\rangle}, 1^{\langle Sz\rangle}, \bot^B} \quad \dfrac{\overline{\vdash 1}^{1}}{\vdash 1, \bot}\bot}{\vdash (1^{\langle Sx\rangle} \otimes \bot), 1^{\langle Sy\rangle}, 1^{\langle Sz\rangle}, 1, \bot^B}\otimes \quad \dfrac{\overline{\vdash 1}^{1}}{\vdash 1, \bot}\bot}{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), 1^{\langle Sz\rangle}, 1, 1, \bot^B}\otimes \quad \dfrac{\overline{\vdash 1}^{1}}{\vdash 1, \bot}\bot}{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), (1^{\langle Sz\rangle} \otimes \bot), 1, 1, 1, \bot^B}\otimes$$

$$\dfrac{\dfrac{\dfrac{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), (1^{\langle Sz\rangle} \otimes \bot), 1, 1, 1, \bot^B}{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), (1^{\langle Sz\rangle} \otimes \bot), 1^{\langle 2\rangle}, 1, \bot^B}\wp}{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), (1^{\langle Sz\rangle} \otimes \bot), 1^{\langle 3\rangle}, \bot^B}\wp}{\vdash (1^{\langle Sx\rangle} \otimes \bot), (1^{\langle Sy\rangle} \otimes \bot), (1^{\langle Sz\rangle} \otimes \bot), (1^{\langle 3\rangle} \wp \bot^B)}\wp$$

The elided proof of $\vdash 1^{\langle Sx\rangle}, 1^{\langle Sy\rangle}, 1^{\langle Sz\rangle}, \bot^B$ is guaranteed to exist by the conditions on the solution to 432-Partition. That is, since x, y, and z are from the same partition, the sum of Sx, Sy, and Sz must be equal to B.

Given the m proofs constructed as above from each of the m groups of elements, one combines them with $\otimes$ into a proof of

$$\vdash (1^{\langle S1\rangle} \otimes \bot), \cdots, (1^{\langle S3m\rangle} \otimes \bot), (1^3 \wp \bot^B)^m$$

The proof can then be completed with $3m$ applications of $\wp$. ∎

2.6 Completeness

Lemma 2.3 (Completeness) *For A, m, B, and S satisfying the constraints of 432-Partition, if there is a proof of the* COMLL *formula $\Theta(\langle A, m, B, S\rangle)$, then the 432-Partition problem $\langle A, m, B, S\rangle$ is solvable.*

Proof.

To simplify this direction of the proof, we use the extra assumption that the "bin size" B is greater than 8. For a justification of this assumption, see Section 2.1. The following makes heavy use of Lemma 2.1.

Assuming we have a proof of

$$\vdash (1^{\langle S1\rangle} \otimes \bot) \wp (1^{\langle S2\rangle} \otimes \bot) \wp \cdots \wp (1^{\langle S3m\rangle} \otimes \bot) \wp (\bot^B \wp 1^{\langle 3\rangle})^m$$

we show that the corresponding 432-Partition problem is solvable.

If there is a proof of this sequent, then there is a cut-free proof, by the cut elimination theorem (Theorem 1.1). By repeated applications of

2.4 Constant-only Encoding

We will now describe how 432-Partition instances (which are at the same time 3-Partition instances) can be encoded in COMLL.

We will use the following notation: $x^Y = \overbrace{x \otimes x \otimes \cdots \otimes x \otimes x}^{Y \text{ copies}}$, as before, and $x^{\langle Y \rangle} = \overbrace{x \wp x \wp \cdots \wp x \wp x}^{Y \text{ copies}}$. Note that $(x^Y)^\perp = (x^\perp)^{\langle Y \rangle}$ and $(x^{\langle Y \rangle})^\perp = (x^\perp)^Y$.

Given an instance of 432-Partition equipped with a set $A = \{a_1, \cdots, a_{3m}\}$, an integer B, and a unary function S, presented as a tuple $\langle A, m, B, S \rangle$, we define the encoding function Θ as $\Theta(\langle A, m, B, S \rangle) =$

$$[(\perp \multimap \perp^{S1}) \otimes \cdots \otimes (\perp \multimap \perp^{S3m})] \multimap [(\perp^3 \multimap \perp^B)^m]$$

Using the contrapositive ($A \multimap B \quad \equiv \quad B^\perp \multimap A^\perp$), we can develop a "1 only" encoding:

$$[(1^{\langle S1 \rangle} \multimap 1) \otimes (1^{\langle S2 \rangle} \multimap 1) \otimes \cdots \otimes (1^{\langle S3m \rangle} \multimap 1)] \multimap [(1^{\langle B \rangle} \multimap 1^{\langle 3 \rangle})^m]$$

Eliminating the linear implication in favor of $\wp$ these formulas both become:

$$(1^{\langle S1 \rangle} \otimes \perp) \wp (1^{\langle S2 \rangle} \otimes \perp) \wp \cdots \wp (1^{\langle S3m \rangle} \otimes \perp) \wp (\perp^B \wp 1^{\langle 3 \rangle})^m$$

We will use the last form of this formula, since it contains no implicit negations (linear implication). One may see this formula satisfies Girard's measure condition, Lemma 2.1, if there are $3m$ elements, and the sum of the sizes equals mB, side conditions on the statement of 432-Partition (and 3-Partition).

The claim is that these formulas are provable in the multiplicative fragment of linear logic if and only if the 432-Partition problem is solvable.

2.5 Soundness

Lemma 2.2 (Soundness) *If a 432-Partition problem $\langle A, m, B, S \rangle$ is solvable, then we are able to find a proof of the* COMLL *formula $\Theta(\langle A, m, B, S \rangle)$.*

Proof.

The proof is straightforward. For each group of three elements in the assumed solution to the 432-Partition problem, one forms the following subproof, assuming the elements of the group are numbered x, y, and z.

Note that if $n = 2$, we have by the above constraint that $X1+X2 = B-1$, and if $n = 4$, then $X1 + X2 + X3 + X4 = B + 1$. Since there are exactly $3m$ elements, and $\sum_{a \in A} s(a) = mB$, there are exactly the same number of groups with four elements as there are groups with two elements.

Further, we may analyze by cases to show that if there are any groups of four, then $B = 4C + 3$ for some integer C. If there are any groups of four, and $B = 4C$ for some C, then the smallest allowable element is $C + 1$, since the size of each element must strictly dominate $B/4$. However, taking four elements of size $C + 1$, the constraint $X1 + X2 + X3 + X4 = B + 1$ is violated. Similarly for $B = 4C + 1$ and $B = 4C + 2$. Thus if there is a group of four elements, then $B = 4C + 3$ for some C, and by simple algebra, the elements of any group of four elements all have size $C + 1$, and the elements of any group of two elements both have size $2C + 1$. Noting that there are exactly as many groups of two as groups of four, we may rearrange the elements of a group of four and a group of two into two groups of three by taking two elements from the group of four and one element from the group of two to form each group of three. Both resulting groups of three have total size $4C + 3$, which happily is equal to B. This "reshuffling" will result in a solution to the 3-Partition problem with the same instance. Therefore 3-Partition and 432-Partition are equivalent problems.

Note that since 432-Partition and 3-Partition are equivalent, 432-Partition is NP-complete in the strong sense. Thus 432-Partition is NP-complete even in unary notation. This is important, since we utilize a unary representation of instances in our linear encoding.

2.3 Encoding with Propositions

We use the notation, for k and c propositions, $x^Y = \overbrace{x \otimes x \otimes \cdots \otimes x \otimes x}^{Y \text{ copies}}$.

Given an instance of 3-Partition equipped with a set $A = \{a_1, \cdots, a_{3m}\}$, an integer B, and a unary function S, presented as a tuple $\langle A, m, B, S\rangle$, we define the encoding function θ as $\theta(\langle A, m, B, S\rangle) =$

$$[(k \multimap c^{S1}) \otimes \cdots \otimes (k \multimap c^{S3m})] \multimap (k^3 \multimap c^B)^m$$

As before, we are writing $S1$ for $s(a_1)$ to improve readability.

It has been show that this formula is provable in the multiplicative fragment of linear logic if and only if the 3-Partition problem is solvable [16].

The encoding using only constants can be generated from this one by replacing k and c by $\perp$.

counting algorithm thus solves this case in polynomial time. Thus for all cases where B is less than or equal to 8 the 3-Partition problem is solvable in polynomial time, and thus 3-Partition remains NP-complete with the further constraint that $B > 8$.

2.2 432-Partition

We introduce a new NP-complete problem, a variant of 3-Partition, which we call 432-Partition:

Instance:	Set A of $3m$ elements, a bound $B \in Z^+$, and a size $s(a) \in Z^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$
Question:	Can A be partitioned into m disjoint sets $A_1, A_2, \cdots, A_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B + \lvert A_i \rvert - 3$?
Comment:	NP-complete in the strong sense.

We will write $S1$ for $s(a_1)$ to improve readability of the following discussion.

We will show that solutions of 432-Partition correspond to solutions of 3-Partition for the same problem instance, under the assumption that $B > 8$. There is a very strong equivalence between these two problems: the instances are the same, instances are solvable in one case exactly when they are solvable in the other, and solutions in one case directly correspond to solutions in the other case. It is clear that solutions to 3-Partition are solutions for the same instance of 432-Partition.

For an arbitrary A_i, let A_i consist of $X1, \ldots Xn$.

If $n = 0$, we have $0 = B - 3$, which is false by our assumption that $B > 8$. If $n = 1$, we have $X1 = B - 2$, but the sizes are bounded above by $B/2$, and with the assumption that $B > 8$, there is a contradiction. Also, considering cases of $n > 4$, we have $\sum_{1 \leq i \leq n} Xi = B + n - 3$, and the assumptions that $B > 8$ and $X_i > B/4$, thus we have $n(B/4) < B + n - 3$, which implies that $n - 3 > B((n/4) - 1)$ and from this and $B > 8$, we have $n < 5$. This leaves the $n = 2$, $n = 3$, and $n = 4$ cases.

Thus we have a partition each element of which consists of either two, three, or four elements.

In the case that $n = 3$, we have $\sum_{1 \leq i \leq 3} Xi = B$, and thus this set identifies a partition which directly satisfies the requirement for 3-Partition, that is, the sum is equal to B.

The main idea is that the small-proof property of MLL allows us to encode "resource distribution" problems naturally. Since linear logic treats propositions as resources natively, it has been called "resource-consciousness" [5]. Note that since full linear logic is conservative over MLL, our encoding remains sound and complete even in larger fragments. This does not lead to new results, however, since the complexity of most larger linear logics have already been completely characterized [15].

2.1 3-Partition

We use the NP-completeness of 3-Partition:
(as stated in Garey+Johnson [6] page 224)

Instance:	Set A of $3m$ elements, a bound $B \in Z^+$, and a size $s(a) \in Z^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$
Question:	Can A be partitioned into m disjoint sets $A_1, A_2, \cdots, A_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B$ (note that each A_i must therefore contain exactly 3 elements from A)?
Reference:	[Garey+Johnson [6], 1975].
Comment:	NP-complete in the strong sense.

Note that 3-Partition is NP-complete in the strong sense, which implies that even when the input is represented in unary, the problem is NP-hard. This property of 3-Partition is essential for our application, since we represent the input problem in unary by multiplicities of linear formulas.

To simplify later arguments we will want to assume that $B > 8$. However there is no loss of complexity with this assumption. One may consider only those instances of 3-Partition where $B > 8$. One may show this by cases. If $B = 0$, or $B = 1$, $B = 2$, or $B = 4$ there are no possible problem instances satisfying $B/4 < s(a) < B/2$. For $B = 5$, all elements must be equal to 2, and thus there are no possible solutions. For $B = 8$, all elements must be equal to 3, and thus there are also no possible solutions in this case. For $B = 3$, all allowable problem instances have all elements equal to 1, and thus this case is solvable in polynomial (constant) time (report "YES"). For $B = 6$, similarly, all elements have size 2, and the answer is trivially "YES". For $B = 7$, all elements have size 2 or 3, and thus all partitions must be made up of two elements of size 2 and one element of size 3. The obvious

The corresponding fact for $\otimes$ does not hold, as demonstrated by the following example $\vdash (1 \parr 1), (\bot \otimes \bot)$.

2 COMLL is NP-complete

Some time ago, Girard [8] developed a necessary condition for the provability of constant multiplicative linear expressions:

Lemma 2.1 (Girard) *Define a function M from constant multiplicative linear expressions to the integers as follows:*

$$\begin{array}{rcl} M(1) & = & 1 \\ M(\bot) & = & 0 \\ M(A \parr B) & = & M(A) + M(B) \\ M(A \otimes B) & = & M(A) + M(B) - 1 \end{array}$$

If a formula A is provable in multiplicative linear logic and contains no propositions, then $M(A) = 1$.

In other words, the number of tensors is one less than the number of ones in any provable COMLL formula. Avron (and others) have studied generalizations of this "semantic" measure to include propositions (where a proposition p is given value 1, and $p^\bot$ is given value 0) yielding a necessary condition for MLL provability. One may go even further, achieving a necessary condition for MALL provability, using min for $\&$ and max for $\oplus$, and plus and minus infinity for the additive constants. For the latter case, the condition becomes: if a formula A is provable in MALL, then $M(A) \geq 1$. Also, one may generalize these conditions somewhat, replacing all instances of 1 with any arbitrary constant c, and allowing propositions to have different (although fixed) values, where p has value v_p, and $p^\bot$ has value $c - v_p$ [3]. Other related work is given in [17] and [4].

Since the above is only a necessary condition, there has been a question as to whether some form of simple "truth table" or numerical evaluation function like the above could yield a necessary and sufficient condition for provability of constant multiplicative (COMLL) expressions. The main result of this paper shows that even this multiplicative constant evaluation or circuit evaluation problem is NP-complete.

We will encode 432-Partition, an NP-complete problem which is a variant of 3-Partition, in MLL, and show that our encoding is sound and complete.

argument for the NP-hardness of this fragment was first sketched by Max Kanovich in electronic mail [10]. Together with the earlier result [15] that the multiplicatives are in NP, Kanovich's result showed that this decision problem is NP-complete. Kanovich later updated his argument to show that the "Horn fragment" of the multiplicatives is also NP-complete [11, 12], using a novel computational interpretation of this fragment of linear logic. This paper continues this trend by providing a proof that evaluating expressions in true, false, and, and or in multiplicative linear logic is NP-complete. That is, even without propositions, multiplicative linear logic is NP-complete.

MLL and COMLL are in NP. Informally, the argument showing membership in NP is simply that every connective in a multiplicative linear logic formula is analyzed exactly once in any cut-free proof. Thus an entire proof, if one exists, can be guessed and checked in nondeterministic polynomial time.

Formally, we first state a fundamental theorem originally due to Girard [7], but proven in complete gory detail in [15].

Theorem 1.1 (Cut Elimination) *If a sequent is provable in* MLL*, then it is provable in* MLL *without using the* **Cut** *rule.*

The above references actually prove this theorem for full linear logic, but the results for the fragments in question here follow immediately. Without cut, multiplicative proofs are quite concise.

Theorem 1.2 (Small-Proofs) *Every connective is analyzed exactly once in any cut-free* MLL *or* COMLL *proof.*

From Theorem 1.1 and Theorem 1.2, we know that given a MLL or COMLL sequent of size n, if there is any proof of this sequent, then there is a proof with exactly n total applications of inference rules. Since each application of an inference rule may be represented in space linear in n, we may simply guess and check an entire n^2 representation of a proof tree in nondeterministic polynomial time.

The following is one of a large family of permutabilities of inferences. Propositional classical logic allows all possible permutabilities (that is, it never matters which formula one choses to break first in a classical proof), and intuitionistic logic exhibits a few impermutabilities [13]. The following permutability of (multiplicative) disjunction holds in linear logic.

Lemma 1.3 (Permutability of $\wp$) *If there is a proof of* $\vdash \Gamma, (A \wp B)$*, then there is a proof of* $\vdash \Gamma, A, B$*.*

Linear negation is defined as follows:

$$\begin{array}{rcl}
(p_i)^\perp & \triangleq & p_i^\perp \\
(p_i^\perp)^\perp & \triangleq & p_i \\
(A \otimes B)^\perp & \triangleq & B^\perp \wp A^\perp \\
(A \wp B)^\perp & \triangleq & B^\perp \otimes A^\perp \\
(A \oplus B)^\perp & \triangleq & A^\perp \,\&\, B^\perp \\
(A \,\&\, B)^\perp & \triangleq & A^\perp \oplus B^\perp \\
(!A)^\perp & \triangleq & ?A^\perp \\
(?A)^\perp & \triangleq & !A^\perp \\
(1)^\perp & \triangleq & \perp \\
(\perp)^\perp & \triangleq & 1 \\
(0)^\perp & \triangleq & \top \\
(\top)^\perp & \triangleq & 0
\end{array}$$

Linear implication, $\multimap$, is defined as follows:

$$A \multimap B \triangleq A^\perp \wp B$$

1.2 Multiplicative Linear Logic

The multiplicative fragment of linear logic MLL is defined as follows. The sequent rules for MLL are the same as those for LL except that the rules for the additive connectives, additive constants, and exponentials are thrown out: $\oplus$, $\&$, ?**W**, ?**C**, ?**D**, !**S**, and $\top$. This leaves only the rules **I**, **Cut**, $\otimes$, $\wp$, $\perp$, and **1**.

1.3 Constant-Only Multiplicative Linear Logic

In this paper, we are concerned with the constant-only multiplicative fragment of linear logic COMLL. The sequent rules for COMLL are those of MLL except **I**. Thus no formulas containing any propositional symbols are provable in COMLL.

1.4 Multiplicative Linear Logic is NP-Complete

In this section we summarize results about the decision problem for propositional multiplicative linear logic which is known to be NP-complete. An

I	$\dfrac{}{\vdash p_i, p_i^{\perp}}$	identity
Cut	$\dfrac{\vdash \Sigma, A \qquad \vdash \Gamma, A^{\perp}}{\vdash \Sigma, \Gamma}$	cut
$\otimes$	$\dfrac{\vdash \Sigma, A \qquad \vdash B, \Gamma}{\vdash \Sigma, (A \otimes B), \Gamma}$	tensor
$\wp$	$\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, (A \wp B)}$	par
$\oplus$	$\dfrac{\vdash \Sigma, A}{\vdash \Sigma, (A \oplus B)} \quad \dfrac{\vdash \Sigma, B}{\vdash \Sigma, (A \oplus B)}$	plus
&	$\dfrac{\vdash \Sigma, A \qquad \vdash \Sigma, B}{\vdash \Sigma, (A \,\&\, B)}$	with
?**W**	$\dfrac{\vdash \Sigma}{\vdash \Sigma, ?A}$	weakening
?**C**	$\dfrac{\vdash \Sigma, ?A, ?A}{\vdash \Sigma, ?A}$	contraction
?**D**	$\dfrac{\vdash \Sigma, A}{\vdash \Sigma, ?A}$	dereliction
!S	$\dfrac{\vdash ?\Sigma, A}{\vdash ?\Sigma, !A}$	storage
$\perp$	$\dfrac{\vdash \Sigma}{\vdash \Sigma, \perp}$	bottom
1	$\dfrac{}{\vdash 1}$	one
$\top$	$\dfrac{}{\vdash \Sigma, \top}$	top

decision problem as simply evaluating expressions in *true*, *false*, *and*, and *or* in multiplicative linear logic ($\otimes$, $\parr$, 1, and $\bot$).

1.1 Propositional Linear Logic

The formal framework we will work with throughout this paper is a Gentzen-style sequent calculus. We discuss three independent logics here: LL (full propositional linear logic), MLL (LL restricted to multiplicative connectives and constants), and COMLL (The constant-only fragment of MLL). We begin with a definition of LL.

A linear logic sequent is a $\vdash$ followed by a multiset of linear logic formulas. Note that in standard presentations of sequent calculi, sequents are often built from sets of formulas, where here we use multisets. This difference is crucial. We assume a set of propositions p_i given, along with their associated negations, $p_i^\bot$. Below we give the inference rules for the linear sequent calculus, along with the definition of negation and implication. The reader should note that negation is a defined concept, not an operator.

The following notational conventions are followed throughout this paper:

p_i	Positive propositional literal
$p_i^\bot$	Negative propositional literal
A, B, C	Arbitrary formulas
Σ, Γ, Δ	Arbitrary multisets of formulas

Thus the identity rule (**I** below) is restricted to atomic formulas, although in fact the identity rule for arbitrary formulas ($\vdash A, A^\bot$) is derivable in this system. For notational convenience, it is usually assumed that $\multimap$ and $\otimes$ associate to the right, and that $\otimes$ has higher precedence than $\multimap$. The notation $?\Sigma$ is used to denote a multiset of formulas which all begin with ?. The English names for the rules given below are are shown on the right. Note that there is no rule for the 0 constant.

(copying) and weakening (throwing away) for propositions. Without contraction or weakening, propositions may be thought of as resources, which must be carefully accounted for. When propositions are treated as resources, as they are in linear logic, one is naturally led to consider two different forms of conjunction and disjunction. Girard named the two kinds of connectives "additive" and "multiplicative", and focussed his attention on the multiplicative fragment by giving proof nets (a version of natural deduction tailored for linear logic) for this fragment. Since then much of the interest in linear logic has revolved around this fragment and small extensions to this fragment.

In order to explain the intuitive difference between additive and multiplicative connectives, consider the conjunctive goal $\Sigma \vdash A \; and \; B$. In all sequent calculi, one must prove $\Delta \vdash A$ and one must also prove $\Gamma \vdash B$, for some Δ and Γ in order to prove this goal. Various sequent calculi place different requirements on the relationship between Σ, Δ, and Γ. For example, in classical logic the latter two are required to be subsets of the first ($\Delta \subseteq \Sigma$ and $\Gamma \subseteq \Sigma$). This may be seen as implicitly allowing copying of some propositions, (those which appear in all three contexts), and throwing away others (those which appear in the conclusion Σ, but not in either hypothesis). The multiplicative conjunction $\otimes$ of linear logic requires that the context Σ be divided between its hypotheses ($\Delta \bigcup \Gamma = \Sigma$ and $\Delta \bigcap \Gamma = \emptyset$). The additive conjunction &, on the other hand, requires that the context be duplicated in both hypotheses ($\Delta = \Gamma = \Sigma$). This critical difference is also reflected in the two forms of disjunction, which are the De Morgan duals of the two forms of conjunction.

Girard also added "exponential" unary connectives to linear logic, increasing the expressive power of the logic greatly. In fact, propositional linear logic with exponentials is undecidable [15]. Without exponentials, Multiplicative-Additive Linear Logic (MALL) is decidable, and is PSPACE-complete [15].

In this paper we focus on the smaller fragment with only the multiplicative connectives and constants, Constant-Only Multiplicative Linear Logic. In an earlier paper, the first author showed that the decision problem for Multiplicative Linear Logic (with propositions) MLL is in NP, by giving (a sketch of) an NP algorithm [15]. However, the NP-hardness of this problem was left open.

Here we show that not only is MLL NP-complete, but the fragment containing no propostions, COMLL is NP-complete as well. Note that this fragment contains no quantifiers or propositions, and thus one may view this

Constant-Only Multiplicative Linear Logic is NP-Complete

Patrick Lincoln*
Computer Science Department
Stanford University
Stanford, CA, 94305
lincoln@cs.stanford.edu

Timothy Winkler
Computer Science Laboratory
SRI International
Menlo Park, CA
winkler@csl.sri.com

August 31, 1992

Abstract

Linear logic is a resource-aware logic that is based on an analysis of the classical proof rules of contraction (copying) and weakening (throwing away). In this paper we study the decision problem for the multiplicative fragment of linear logic without quantifiers or propositions: the constant-only case. We show that this fragment is NP*-complete. Earlier work by Max Kanovich showed that propositional multiplicative linear logic is* NP*-complete. With Natarajan Shankar, the first author developed a simplified proof for the propositional case. The structure of this simplified proof is utilized here with a new encoding which uses only constants. The end product is the somewhat surprising result that simply evaluating expressions in* true, false, and, *and* or *in multiplicative linear logic (*$\otimes$, $\parr$, 1, *and* $\bot$*) is* NP*-complete. By conservativity results not proven here, the* NP*-hardness of larger fragments of linear logic follows.*

1 Introduction

When Girard introduced linear logic [7], he brought to light the expressive power which can be gained by restricting the structural rules of contraction

*Supported by AT&T Doctoral Scholarship and SRI.